XIDENT - A COMPUTER TECHNIQUE FOR THE DIRECT INDEXING OF ELECTRON DIFFRACTION SPOT PATTERNS

B. L. RHOADES

Department of Mechanical Engineering,
University of Canterbury, Christchurch,
New Zealand.

Research Report 70/76

C O N T E N T S

ABSTRACT

A rapid computer technique to index electron diffraction spot patterns from any crystal structure is described. The diffracting zone, which is represented to the computer by the measured diffracted distances of three or five diffraction spots and the angles between them, is compared with the reciprocal lattices of phases likely to be present, and all zones which correspond within prescribed limits are printed out in order of best fit. The program provides for automatic rejection of prohibited reflections for the Bravais lattices, and for the suppression of symmetrical solutions if desired. The technique is of general application, but has proved particularly useful for the indexing of patterns from unidentified non-cubic structures in situations where experimental error must be tolerated.

INTRODUCTION

XIDENT is a FORTRAN IV computer program which has been developed to directly index electron diffraction spot patterns from any crystal structure using a computer of at least 64K words of core storage capacity. A rapid identification technique is used which enables a large number of prospective structures to be examined while accommodating the experimental error which is inherent in selected area electron diffraction using an electron microscope. The speed of the program is such that all possible orientations of each phase are considered, and every reciprocal lattice plane which corresponds to the diffraction pattern, within the prescribed limits of accuracy, will be identified.

An outline flow diagram of XIDENT is shown in Figure 1, and a FORTRAN listing, which contains detailed input specifications, is attached. Figure 2 is a typical diffraction pattern from the tip of an iron oxide whisker crystal and will be used to illustrate a typical application of the program.

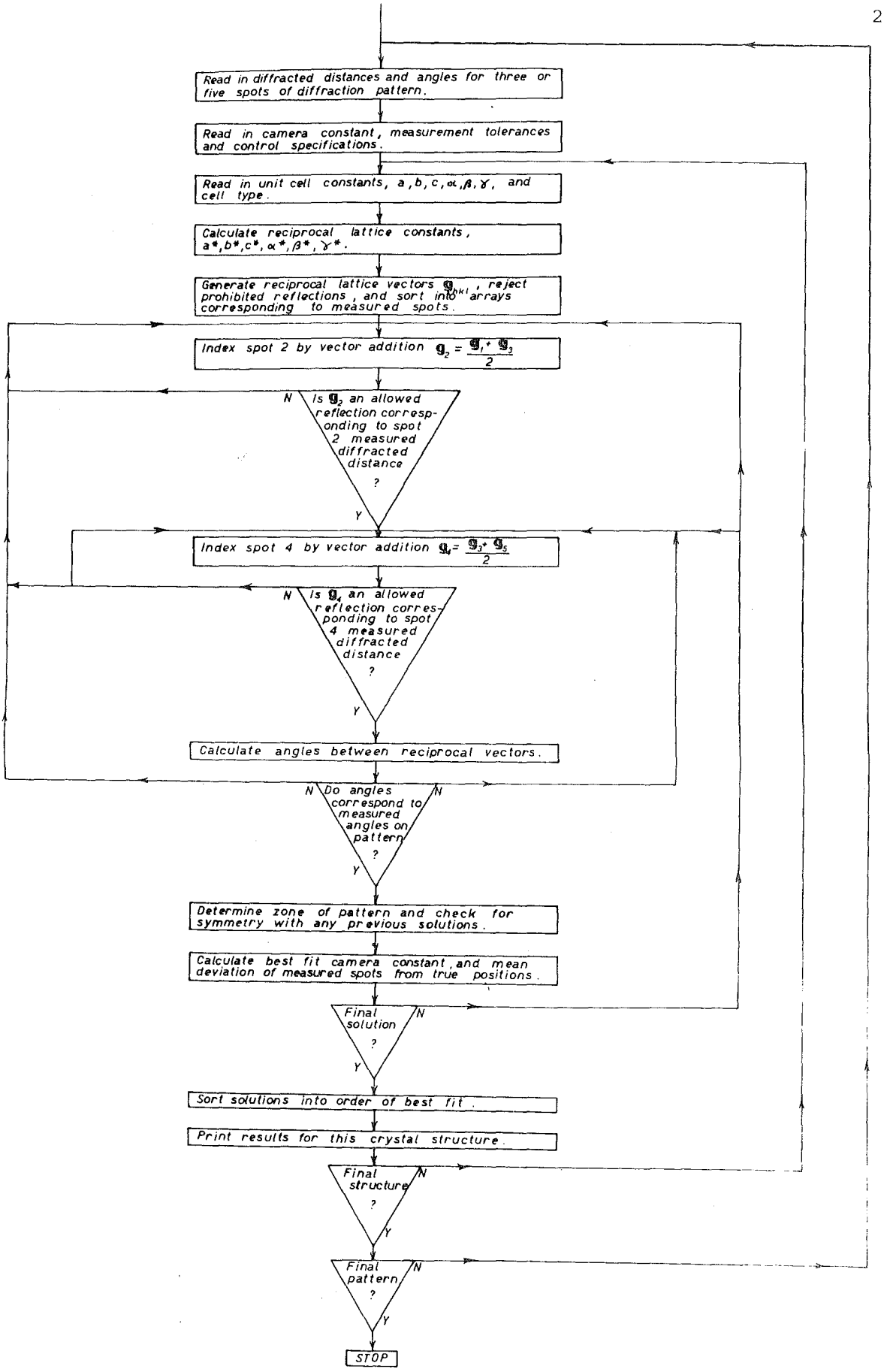
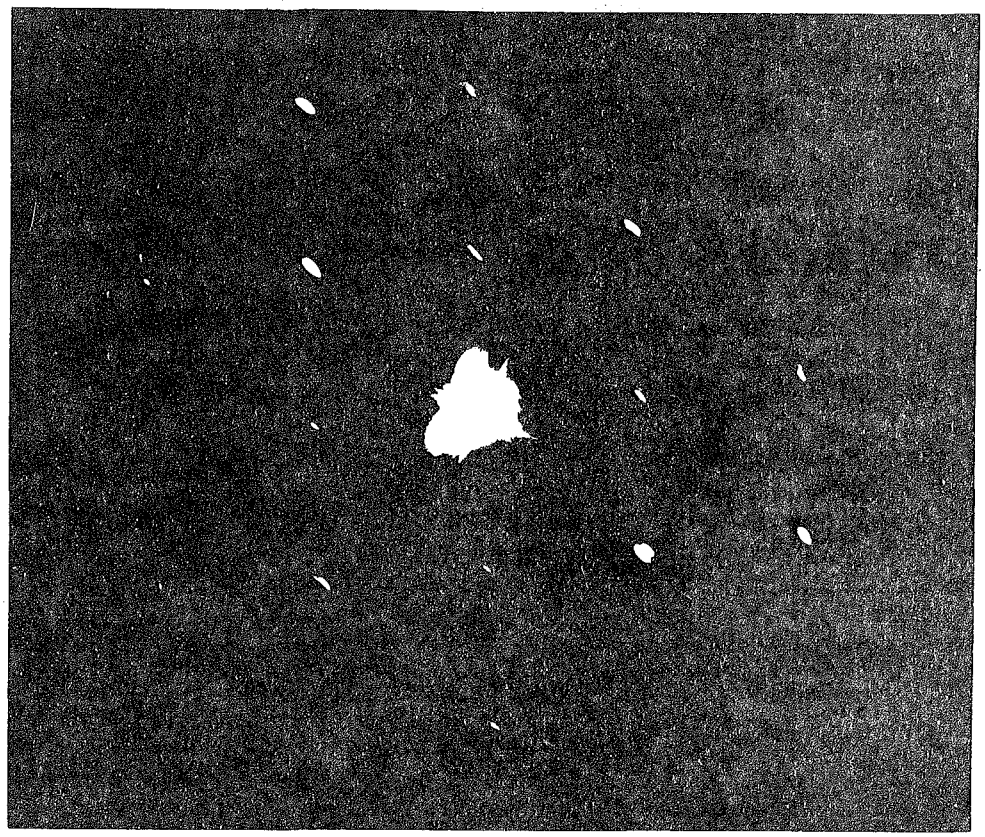


Figure 1. Outline flow diagram of program XIDENT.



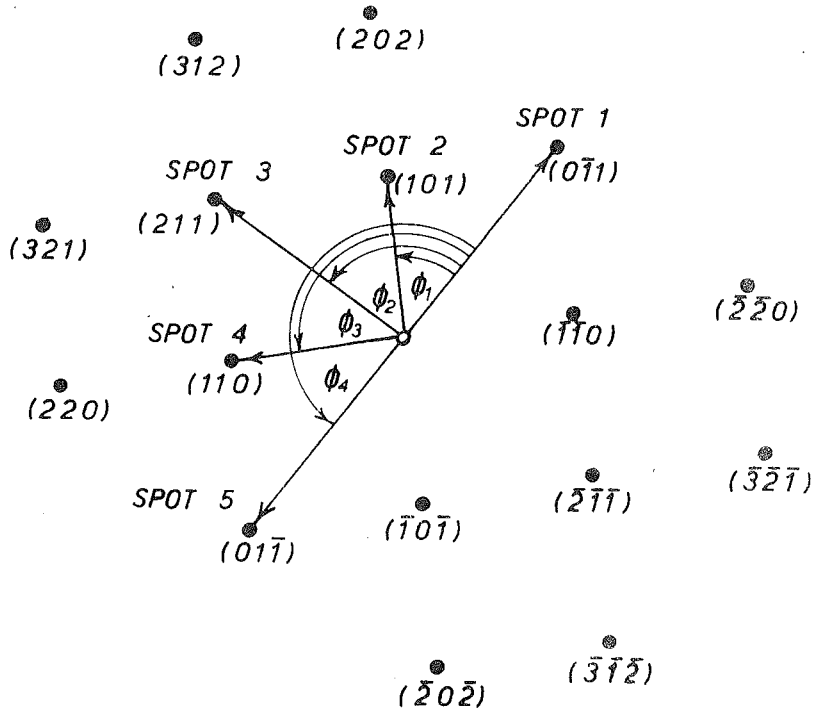


Figure 2. Selected area electron diffraction pattern from αFe_2O_3 whisker. The $[\bar{1}11]$ zone (structural rhombohedral cell) is shown, with streaks perpendicular to $[0\bar{1}1]$ whisker growth direction.

PROGRAM XIDENT

1 Data Input

The reciprocal lattice net of the spot pattern is represented to the computer by the input of the diffracted distances of either three or five diffraction spots, an approximate camera constant, and the measured angles ϕ_1, ϕ_2, etc. between these spots. The spots are chosen in groups of three to represent rows of the reciprocal lattice; spots 1, 2 and 3 being three equi-spaced points in any non-central row of the diffraction pattern, and spots 4 and 5 being chosen in conjunction with spot 3 to represent another non-central row in a similar manner. Figure 2 illustrates the choice of five spots according to this criterion, and it will be seen that the geometry of a single zone diffraction pattern may be completely characterised by these five spots except for the positions of possible forbidden reflections. The innermost spots on the pattern are normally used as input data since these represent the crystal lattice planes of greatest separation. These planes are fewer in number and are more characteristic of a particular structure than are the planes represented by spots further from the centre. It is not essential to choose the innermost spots, however, as these may frequently be obscured by a superimposed pattern from a diffraction standard or from a specimen support film, but computation times are lower and there is less likelihood of ambiguous results when the innermost spots are chosen.

Any three equi-spaced spots in one non-central row of the pattern may be chosen as an alternative form of input; spots 1, 2 and 3 from Figure 2 being acceptable input. It is plain from the figure that such an input may not completely describe the two-dimensional nature of a diffraction net and, although XIDENT will index the pattern normally, there is a possibility that incorrect solutions will be produced. This form of input may be useful for the indexing of imperfect patterns which

are produced when the zone axis of the pattern is inclined to the electron beam but it is generally preferable to use five spots to avoid erroneous solutions.

Experimental error is accommodated by the input of three tolerances which relate to the following:

(a) errors in the camera constant (expressed as a percentage);

(b) errors in measurement of the diffracted distances of individual spots (expressed in millimetres); and

(c) errors in the measured interplanar angles from the diffraction pattern (expressed in degrees).

It is necessary to separate tolerances (a) and (b). Errors in the determination of the camera constant; owing to variation of objective lens current, specimen position, or other instrument factors; tend to dominate for spots with large diffracted distances; whereas errors in individual spot positions, attributable to the presence of reciprocal lattice streaks inclined to the direction of the electron beam and other factors which lead to poor measurement precision, are of greater significance for spots nearer the centre of the pattern.

2 Generation of Reciprocal Vectors

Indexing of the five diffraction spots is accomplished by matching of the measured diffracted distances and interplanar angles to the magnitudes of vectors $\vec{g}_{hkl}$ in the reciprocal lattice and the angles ϕ_1, ϕ_2, etc. between these vectors. Reciprocal vectors are generated for all possible combinations of Miller indices by successive decrementing of the values of (hkl) from (nnn) to $(\bar{n}\bar{n}\bar{n})$, where the value of n is chosen so that only vectors with magnitudes small enough to correspond with one of the input diffracted distances will be generated.

To determine the magnitude of the reciprocal vectors it is first necessary to define the reciprocal unit cell in terms of the unit cell

vectors $\vec{a}$, $\vec{b}$ and $\vec{c}$ for the crystal being considered. The reciprocal lattice unit cell is defined by the vectors $\vec{a}^*$, $\vec{b}^*$, $\vec{c}^*$, where $\vec{a}^*$ is the axis normal to the plane containing $\vec{b}$ and $\vec{c}$, the $\vec{b}^*$ axis is normal to the plane containing $\vec{c}$ and $\vec{a}$, and the $\vec{c}^*$ axis is normal to the plane containing $\vec{a}$ and $\vec{b}$.

$$\text{hence} \quad \vec{a}^*.\vec{a} = \vec{b}^*.\vec{b} = \vec{c}^*.\vec{c} = 1 \qquad \ldots.(1)$$

$$\text{and} \quad \vec{a}^*.\vec{b} = \vec{b}^*.\vec{c} = \vec{c}^*.\vec{a} = \vec{a}^*.\vec{c} = 0 \qquad \ldots.(2)$$

The reciprocal vectors can thus be determined from the direct cell vectors by the relationships

$$\vec{a}^* = \frac{\vec{b} \times \vec{c}}{V}; \quad \vec{b}^* = \frac{\vec{c} \times \vec{a}}{V}; \quad \vec{c}^* = \frac{\vec{a} \times \vec{b}}{V} \qquad \ldots.(3)$$

where V is the volume of the direct unit cell bounded by a, b, and c and is given by the triple product,

$$V = \vec{a}.(\vec{b} \times \vec{c})$$

$$= abc\,(1 + 2\cos\alpha\cos\beta\cos\gamma - \cos^2\alpha - \cos^2\beta - \cos^2\gamma)^{\frac{1}{2}} \qquad \ldots.(4)$$

where α, β, γ are respectively the angles between

$\vec{b}$ and $\vec{c}$, $\vec{c}$ and $\vec{a}$, and $\vec{a}$ and $\vec{b}$.

The angles α^*, β^* and γ^* between $\vec{b}^*$ and $\vec{c}^*$, $\vec{c}^*$ and $\vec{a}^*$, and $\vec{a}^*$ and $\vec{b}^*$ respectively are given by the scalar product:

$$\cos\alpha^* = \frac{\vec{b}^*.\vec{c}^*}{b^*c^*}, \quad \text{etc.}$$

That is

$$\cos\alpha^* = \frac{\cos\beta\cos\gamma - \cos\alpha}{\sin\beta\sin\gamma}$$

$$\cos\beta^* = \frac{\cos\alpha\cos\gamma - \cos\beta}{\sin\alpha\sin\gamma}$$

$$\cos\gamma^* = \frac{\cos\alpha\cos\beta - \cos\gamma}{\sin\alpha\sin\beta} \qquad \ldots.(5)$$

It can be shown that the reciprocal vector

$$\vec{g}_{hkl} = h\vec{a}^* + k\vec{b}^* + l\vec{c}^* \qquad \ldots.(6)$$

is normal to the crystal lattice plane with Miller indices (hkl), and that

$$|\vec{g}_{hkl}| = \frac{1}{d_{hkl}}$$

where d_{hkl} is the spacing of these lattice planes.

Three vectors $\frac{\vec{a}}{h}$, $\frac{\vec{b}}{k}$, and $\frac{\vec{c}}{l}$ determine the position of the crystal lattice plane (hkl) which passes through their end points A, B and C as shown in Figure 3. Vector $\vec{AB}$ is $\frac{\vec{b}}{k} - \frac{\vec{a}}{h}$, and the scalar products of $\vec{AB}$ and $\vec{g}_{hkl}$ is

$$\left(\frac{\vec{b}}{k} - \frac{\vec{a}}{h}\right).(h\vec{a}^* + k\vec{b}^* + l\vec{c}^*) = 0 \qquad \ldots.(7)$$

from equations (1) and (2). Hence $\vec{g}_{hkl}$ is perpendicular to $\vec{AB}$, and similarly is perpendicular to $\vec{BC}$, $\vec{CA}$, or any vector in this plane; i.e. $\vec{g}_{hkl}$ is perpendicular to the plane (hkl). Also, if d_{hkl} is the length of the normal to this plane from the origin of the coordinates and if $\vec{n}$ is a unit vector along the normal,

$$d_{hkl} = \vec{n}.\frac{\vec{a}}{h} = \frac{(h\vec{a}^* + k\vec{b}^* + l\vec{c}^*)}{|\vec{g}_{hkl}|}.\frac{\vec{a}}{h} = \frac{1}{|\vec{g}_{hkl}|} \qquad \ldots.(8)$$

Therefore $\vec{g}_{hkl}$ is normal to the crystal plane (hkl) and its magnitude $|\vec{g}_{hkl}| = \frac{1}{d_{hkl}}$. Hence

$$\vec{g}.\vec{g} = \frac{1}{d_{hkl}^2} = h^2(\vec{a}^*.\vec{a}^*) + 2hk(\vec{a}^*.\vec{b}^*) + 2hl(\vec{a}^*.\vec{c}^*) + k^2(\vec{b}^*.\vec{b}^*) + 2kl(\vec{b}^*.\vec{c}^*) + l^2(\vec{c}^*.\vec{c}^*) \qquad \ldots.(9)$$

From the properties of the scalar product

$$|\vec{g}_{hkl}| = \frac{1}{d_{hkl}} = (h^2a^{*2} + 2hka^*b^*\cos\gamma^* + 2hla^*c^*\cos\beta^* + k^2b^{*2} + 2klb^*c^*\cos\alpha^* + l^2c^{*2})^{\frac{1}{2}} \qquad \ldots.(10)$$

3 Indexing of Spots by Vector Addition

Vectors with magnitudes falling within the prescribed tolerances for the measured diffracted distances for spots 1, 3 and 5 are sorted into

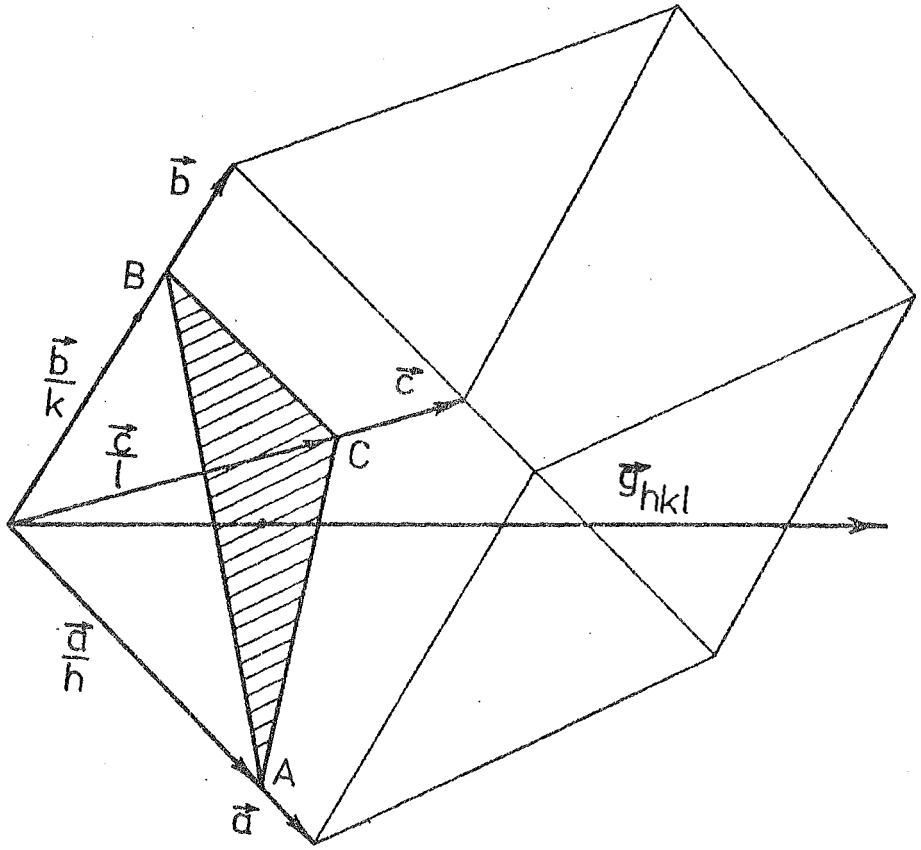


Fig. 3 Relation between the reciprocal lattice vector $\vec{g}_{hkl}$ and the unit cell vectors $\vec{a}$, $\vec{b}$, $\vec{c}$.

three arrays as they are generated. In the listed version of the program provision has been made for up to 800 reciprocal vectors corresponding to each spot. However, this number may be readily altered to match the available core storage on any computer system. In the unusual event that more than 800 reciprocal vectors may be matched to any one diffraction spot, it is necessary to either choose new input data spots nearer to the centre of the pattern or reduce the tolerance on camera constant or measured diffracted distance.

During this loading operation reciprocal vectors corresponding to prohibited reflections for the lattice types listed in Table 1 may be automatically removed. The appropriate control information (SYM) to initiate this procedure is contained in the first column of the data card describing the crystal structure. If forbidden reflections are removed in this way the input data diffraction spots must be chosen with care, as it is possible that two-dimensional diffraction and secondary scattering may mask the extinction laws for a structure and lead to the appearance of forbidden reflections in the diffraction pattern . If a forbidden spot is chosen as input data, and reciprocal vectors for forbidden reflections have been removed, the correct indexing will be automatically rejected by XIDENT.

Indexing of the pattern proceeds by considering the spots in groups of three. All $\vec{g}_{h_1k_1l_1}$ vectors corresponding to point 1 are sequentially matched with $\vec{g}_{h_3k_3l_3}$ vectors corresponding to point 3, and a tentative indexing of point 2 is performed by vector addition:

$$h_2 = \frac{h_1 + h_3}{2}, \quad k_2 = \frac{k_1 + k_3}{2}, \quad l_2 = \frac{l_1 + l_3}{2} \qquad \ldots.(11)$$

If $\vec{g}_{h_2k_2l_2}$ corresponds to an allowed reflection, its magnitude is calculated and compared with the measured diffracted distance for point 2. If $\left|\vec{g}_{h_2k_2l_2}\right|$ falls within the allowed tolerance for spot 2 an attempt is

CLASS OF REFLECTIONS	LATTICE TYPE	LATTICE SYMBOL	FORBIDDEN REFLECTIONS	SYM
hkl	Primitive	P	none	0
"	All faces centred	F	$h + k \neq 2n$ $k + l \neq 2n$ $h + l \neq 2n$	1
"	Body centred	I	$h + k + l \neq 2n$	2
"	A face centred	A	$k + l \neq 2n$	3
"	B face centred	B	$h + l \neq 2n$	4
"	C face centred	C	$h + k \neq 2n$	5
"	Rhombohedral, obverse	R	$-h + k + l \neq 3n$	6
"	Rhombohedral, reverse	R	$h - k + l \neq 3n$	7

TABLE 1 Prohibited reflections removed by subroutine PROHIB.

made to index spot 4 by a similar process. (If three spots only are input the program moves directly to the angles check described in Section 4.2.4.)

$$h_4 = \frac{h_3 + h_5}{2}, \quad k_4 = \frac{k_3 + k_5}{2}, \quad l_4 = \frac{l_3 + l_5}{2}$$

The magnitude of $\vec{g}_{h_4k_4l_4}$ is then compared with the measured diffracted distance for spot 4.

4 Comparison of Interplanar Angles

If five reciprocal vectors are found with magnitudes corresponding to the measured diffracted distances and with hkl values satisfying the vector inter-relationships implied by the geometry of the input data spots, then the angles between these vectors ϕ_1', ϕ_2', etc., are calculated and compared with the angles ϕ_1, ϕ_2, etc., measured from the diffraction pattern.

These angular relationships are determined by consideration of the scalar product of the relevant reciprocal vectors.

$$\vec{g}_1 . \vec{g}_2 = \left|\vec{g}_1\right| \left|\vec{g}_2\right| \cos \phi' \qquad \ldots.(12)$$

therefore

$$\cos \phi' = \frac{(h_1\vec{a}^* + k_1\vec{b}^* + l_1\vec{c}^*).(h_2\vec{a}^* + k_2\vec{b}^* + l_2\vec{c}^*)}{\left|\vec{g}_1\right| \left|\vec{g}_2\right|}$$

$$= [h_1h_2a^{*2} + k_1k_2b^{*2} + l_1l_2c^{*2} + (k_1l_2 + l_1k_2)b^*c^*\cos\alpha^*$$

$$+(l_1h_2 + h_1l_2)c^*a^*\cos\beta^* + (h_1k_2 + k_1h_2)a^*b^*\cos\gamma^*](\left|\vec{g}_1\right| \left|\vec{g}_2\right|)^{-1}$$

$$\ldots.(13)$$

Evaluation of $\left|\vec{g}_1\right|$ and $\left|\vec{g}_2\right|$ from Equation 10 provides the required

angle in terms of the reciprocal cell constants and the Miller indices of the relevant crystal planes.

If all angles correspond with those measured, within the specified tolerance, the indexing of the spots is recognised as a valid solution. The zone axis of the pattern [uvw] is then determined by cross multiplication of $\vec{g}_{h_1k_1l_1}$ and $\vec{g}_{h_2k_2l_2}$ and division by the highest common factor.

$$u = k_1l_2 - k_2l_1 \ , \quad v = l_1h_2 - l_2h_1 \ , \quad w = h_1k_2 - h_2k_1 \qquad \ldots(4.14)$$

No attempt is made to check that spots 3, 4 and 5 lie in the same reciprocal lattice plane as 1 and 2 as XIDENT may index spots across overlapping Laue zones if the vector additive relationships are satisfied.

5 Sorting of Results and Output

If more than one successful indexing is found for a given phase, the indexed solutions are sorted into order of accuracy based on the mean deviation of the measured spots from their true positions which are determined on the basis of a calculated best fit camera constant.

$$C_{\text{best fit}} = C\left[\frac{\sum_{n=1}^{m}\left(d_n - \frac{C}{2r_n}\right)}{\sum_{n=1}^{m}\frac{C}{2r_n}} - 1\right] \qquad \ldots.(15)$$

Where $C_{\text{best fit}}$ is the 'best fit' diametral camera constant, C is the input diametral camera constant; r_n is the measured diffracted distance to spot n ; d_n is the calculated d-spacing corresponding to spot n ; and m is the number of spots being considered (either three or five).

Figure 4 illustrates the procedure for determining the mean deviation of the measured spots from their "true positions". Since spot 1 is the datum for the measurement of angles ϕ_1, ϕ_2, etc., it is necessary

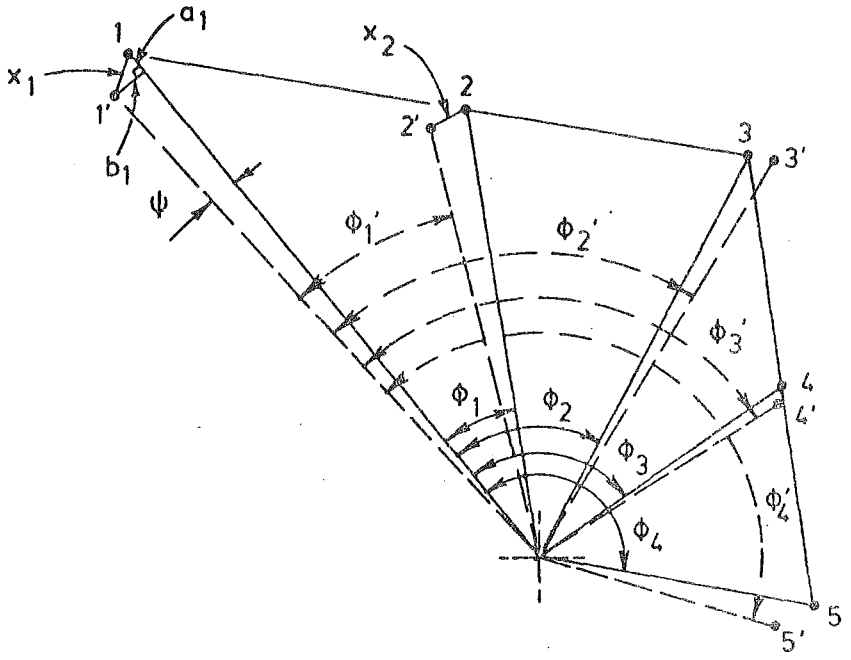


Fig. 4 Determination of mean deviation of diffraction spots from "true positions".

1, 2, 3, etc. are the measured positions of input data spots.1', 2', 3', etc. are calculated "true positions" based on the best fit camera constant.

to make correction for any small initial error in the angular position of spot 1 relative to the other spots; i.e. the pattern is effectively rotated by an angle ψ to obtain the best average correspondence between the measured angles ϕ_1, ϕ_2, etc., and the calculated angles ϕ'_1, ϕ'_2, etc. between the relevant reciprocal vectors. The required rotation is given by:

$$\psi = \frac{-\sum_{n=1}^{m} (\phi'_n - \phi_n)}{m} \qquad \ldots.(16)$$

The mean deviation of the spots from their true positions $\frac{\Sigma x}{m}$ is now calculated with reference to the notation of Figure 4,

$$\frac{\Sigma x}{m} = \frac{\sum_{n=1}^{m} (a_n^2 + b_n^2)^{\frac{1}{2}}}{m} \qquad \ldots.(17)$$

where $$a_n = r_n - \frac{C_{best\ fit} \cdot \cos(\phi'_{n-1} - \phi_{n-1} - \psi)}{2d_n} \qquad \ldots.(18)$$

and $$b_n = \frac{C_{best\ fit} \cdot \sin(\phi'_{n-1} - \phi_{n-1} - \psi)}{2d_n} \qquad \ldots.(19)$$

A sample printout from program XIDENT is shown in Figure 5. This printout corresponds to the diffraction pattern of Figure 2. If more than one set of equivalent solutions is detected, the symmetrical equivalent solutions are grouped together and are printed out in order of accuracy. The printing of equivalent solutions may be suppressed if desired.

The 180° rotation ambiguity which exists for the indexing of each reciprocal lattice plane is resolved during the indexing process so that only one valid indexing is recognised for each zone. This is achieved by the termination of loading of reciprocal vectors into the array corresponding to spot 3 when the (hkl) values are decremented through (000). This expediency halves the number of reciprocal vectors corresponding to

```
PHOTO 5502 IRON OXIDE WHISKER
* * * * * * * * * * * * * * * *

CAMERA CONSTANT = 44.75
POSITION TOLERANCE = 1.00 MILLIMETRES (MINIMUM OVER-RIDING TOLERANCE OF + OR - 5.0 PERCENT OF DIFF.
                     DISTANCE PREVAILS)
DIFFRACTED DISTANCES OF SPOTS ARE
  9.05     6.10     8.40     6.15     9.05 (MILLIMETRES)
ANGLE TOLERANCE = 3.00 DEGREES
MEASURED ANGLES BETWEEN SPOTS ARE
  45.00    90.00   135.00   180.00

ALPHA FE 203 RHOMBOHEDRAL
* * * * * * * * * * * * * *

REAL CELL CONSTANTS

     A        B        C       ALPHA       BETA       GAMMA
  5.427    5.427    5.427    55.260      55.260      55.260

       MAXIMUM INDICES ARE       3  3  3
       MAXIMUM DIMENSIONS OF ARRAYS ARE      20    6    20

SET 1  ZONE AXIS  [-1   1   1]   (SET 1 HAS 6 SYMMETRICAL EQUIVALENT SOLUTIONS)
* * * * * * * * * * * * * * * * * *

    SPOT           PLANE            DSPACE          ESTIMATED DSPACE FROM DIFF.PATTERN
      1         ( 0 -1  1)           2.517                          2.502
      2         ( 1  0  1)           3.682                          3.713
      3         ( 2  1  1)           2.699                          2.696
      4         ( 1  1  0)           3.682                          3.682
      5         ( 0  1 -1)           2.517                          2.502
 N.B.  BEST FIT CAMERA CONSTANT USED IN ABOVE ESTIMATES OF D SPACINGS = 45.294
                                                (INPUT CAMERA CONSTANT  = 44.750)
       MEAN DEVIATION OF MEASURED SPOTS FROM TRUE POSITIONS = 0.110 MILLIMETRES


 ANGLE BETWEEN PLANES  1 & 2 =   43.00 (MEASURED  45.00) DEGREES
 ANGLE BETWEEN PLANES  1 & 3 =   90.00 (MEASURED  90.00) DEGREES
 ANGLE BETWEEN PLANES  1 & 4    137.00 (MEASURED 135.00) DEGREES
 ANGLE BETWEEN PLANES  1 & 5 =  180.00 (MEASURED 180.00) DEGREES

 SYMMETRICAL EQUIVALENT SOLUTIONS FOR SET 1

   ZONE AXIS      POINT 1      POINT 2      POINT 3      POINT 4      POINT 5
 [ 1  1 -1]    (-1  1  0)   ( 0  1  1)   ( 1  1  2)   ( 1  0  1)   ( 1 -1  0)
 [ 1 -1  1]    ( 1  0 -1)   ( 1  1  0)   ( 1  2  1)   ( 0  1  1)   (-1  0  1)
 [ 1 -1 -1]    ( 0  1 -1)   ( 1  1  0)   ( 2  1  1)   ( 1  0  1)   ( 0 -1  1)
 [-1 -1  1]    ( 1 -1  0)   ( 1  0  1)   ( 1  1  2)   ( 0  1  1)   (-1  1  0)
 [-1  1 -1]    (-1  0  1)   ( 0  1  1)   ( 1  2  1)   ( 1  1  0)   ( 1  0 -1)
```

Figure 5. Printout from program XIDENT corresponding to the diffraction pattern of Fig.2. Burroughs computation time for this identification = 1.25 seconds.

this spot as vectors of the type $(00\bar{\ell})$, $(0\bar{k}\ell)$ and $(\bar{h}k\ell)$ are rejected. The result of this action is that duplicate solutions for each zone are eliminated and program execution time is substantially decreased.

6 Application of Program XIDENT

XIDENT has proved a most useful technique for the indexing of most electron diffraction spot patterns, although the primary field of application is envisaged as the indexing of patterns from non-cubic structures when the identity of the diffracting phase is not known. The testing of published spot patterns, when the original identification is in doubt, can also be rapidly accomplished in a similar manner to the example shown. An approximate camera constant may be obtained from the original indexing and a search for alternative identification may then be performed over wide experimental limits.

It will be observed from Figure 5 that wide limits of error have been allowed in the indexing process. A camera constant tolerance of $\pm$ 2% to $\pm$ 5% has been routinely used for diffraction patterns from oxide whisker crystals as it is generally impossible to position a diffraction standard co-planar with specimens of this type. Wide measurement error limits of $\pm$ 1.0 mm on diffracted distance and $\pm$ 3.0° on measured angles have also been routinely utilised to accommodate the displacement of spots resulting from pronounced reciprocal lattice streaks in diffraction patterns from filamentary crystals. Despite these large tolerances most materials can be checked within a few seconds of processing time on the Burroughs B6718 computer system and unambiguous results are generally obtained.

A number of alternative computer indexing techniques have been examined within the basic structure of program XIDENT. The indexing method described in Section 3 has a significant advantage over most other possible methods of vector addition in that it is not necessary to utilise the innermost spots on the diffraction pattern which may

frequently be obscured by a superimposed pattern. Indexing of patterns by the matching of interplanar angles alone, in the manner described by Wilkes[1] and by Booth[2] *et al*, has also been attempted. It was found that the matching of reciprocal vectors and diffraction spots on the basis of an angles comparison was considerably slower than the indexing technique of Section 3 owing to the complexity of the matrix manipulations required to calculate such angles compared to the simple vector addition technique of XIDENT. The absence of any attempt to compare the relative positions of reciprocal lattice vectors and diffraction spots by vector addition also resulted in the production of large numbers of incorrect solutions which required manual verification.

7 SAMPLE DATA DECK

The form of the input data is described at the head of the program listing. The data deck required for the example shown in Figure 5 is as follows :-

```
PHOTObb5502bIRONbOXIDEbWHISKER
bb44•750bl•0001
bbb9•050bbb6•100bbb8•400bbb6•150bbb9•050
bbb45•00bbb90•00bbl35•00bbl80•00
OALPHAbFE203bRHOMBOHEDRAL
bbbbb5•427bbbbb5•427bbbbb5•427bbbbb55•26bbbbb55•26bbbbb55•26
```

REFERENCES

1. WILKES, P. Complete indexing of electron diffraction patterns by computer. J Mater. Sci. 9, 1974: 517.

2. BOOTH, M. *et al*. A general program for interpreting electron diffraction patterns. Metall. Trans. 5, 1974: 775.

```
C      @@        @@  @@@@@  @@@@@@@@@@  @@@@@@@@@@@@  @@        @@  @@@@@@@@@@@@
C      @@@@      @@@@  @@@@  @@@@@@@@@@  @@@@@@@@@@@@  @@@@      @@  @@@@@@@@@@@@
C        @@    @@    @@  @@  @@    @@    @@    @@      @@@@@@    @@    @@
C          @@ @@      @@  @@  @@    @@    @@    @@      @@ @@ @@  @@    @@
C            @@@@     @@  @@  @@    @@    @@    @@      @@  @@@@  @@    @@
C             @@@     @@  @@  @@    @@    @@    @@      @@   @@   @@    @@
C           @@ @@     @@  @@  @@    @@    @@    @@      @@        @@    @@
C         @@    @@    @@  @@  @@    @@    @@    @@      @@        @@    @@
C       @@@@      @@@@  @@@@  @@@@@@@@@@  @@@@@@@@@@@@  @@        @@    @@
C      @@          @@  @@@@  @@@@@@@@@@  @@@@@@@@@@@@  @@        @@    @@
C
C
C      A COMPUTER PROGRAM FOR THE INDEXING OF ELECTRON DIFFRACTION SPOT P ATTERNS
C      BY B.L. RHOADES - DEPARTMENT OF MECHANICAL ENGINEERING, UNIVERSITY  OF
C      CANTERBURY, NEW ZEALAND
C
C
C      FORM OF INPUT DATA
C      **********
C      **********
C
C      DATA CARD 1  DETAILS OF DIFFRACTION PATTERN          FORMAT (20A4)
C      ***********
C
C      DATA CARD 2  CAMCO, TOL, IOUT                        FORMAT (F8.4, F6,3,I1)
C      ***********
C                   CAMCO IS A DIAMETER BASED CAMERA CONSTANT (2*LAMBDA*L ) IN
C                   MILLIMETRE*ANGSTROMS, TOL IS A MEASUREMENT TOLERANCE  IN
C                   MILLIMETRES, IOUT=1 IF PRINTOUT OF SYMMETRICAL SOLUTI ONS IS
C                   REQUIRED OTHERWISE LEAVE BLANK
C
C      DATA CARD 3  DIFFRACTED DISTANCES OF FIVE SPOTS (MM) FORMAT (5F8.3 )
C      ***********
C
C      DATA CARD 4  FOUR ANGLES BETWEEN SPOTS (DEGREES)     FORMAT (4F8.2 )
C      ***********
C
C      DATA CARD 5  SYM, TITLE OF MATERIAL                  FORMAT (I1,19 A4)
C      ***********
C                   SYM IS A CONTROL VARIABLE FOR THE REMOVAL OF  PROHIBI TED
C                   REFLECTIONS          FOR SYM WRITE:
C                         0 FOR PRIMITIVE CELL TYPE P
C                         1 FOR ALL FACES CENTRED TYPE F
C                         2 FOR BODY CENTRED CELL TYPE I
C                         3 FOR A FACE CENTRED CELL TYPE A
C                         4 FOR B FACE CENTRED CELL TYPE B
C                         5 FOR C FACE CENTRED CELL TYPE C
C                         6 FOR OBVERSE RHOMB (HEX CELL) TYPE R
C                         7 FOR REVERSE RHOMB (HEX CELL) TYPE R
C
C      DATA CARD 6  CELL CONSTANTS                          FORMAT(3F10.3 ,3F10.2)
C      ***********
C                   A,B,C IN ANGSTROMS, ALPHA,BETA,GAMMA IN DEGREES
C
C      DATA CARD 7  ICARD                                   FORMAT (I1)
C      ***********
C                   ICARD CONTROLS LOOPING OF THE PROGRAM
C                   IF NEW MATERIAL FOLLOWING   ICARD=1 AND REPEAT CARDS  5-7
C                   IF NEW DIFF. PATTERN FOLLOWING  ICARD=2 AND REPEAT CA RDS 1-7
C
C
C
C
C      PROGRAM XIDENT
C


      INTEGER READR,PRINTR,H,HMAX,PMAX,QMAX,RMAX,SMAX,TMAX,SYM,SYMI
      DOUBLE PRECISION DSQ2,DSQ4
      DOUBLE PRECISION A,B,C,PI180,V,DMAX,DSQ,ASTAR,BSTAR
      DOUBLE PRECISION CSTAR,SINA,SINB,SING,COSA,COSB,COSG,COSAS
      DOUBLE PRECISION COSBS,COSGS,A11,A12,A13,A22,A23,A33,DMIN
      COMMON RADIX(5,50),ANGIX(4,50),IZONES(3,50,50),IHP(5,50,50),
```

```
     1IKP(5,50,50),ILP(5,50,50),KSETS,KP(50),RAD(3,801),KS,ANGLES(4),
     2IZOS(3,50),IOUT,DIST(5),DISTIX(5,50),ANGACT(4),CAMKO(50),CAMCO,
     3LE,LES,ADEVN(50),DEVSQ(50),KT,D2,D4
      DIMENSION TITLE(19),IND(3,3,801),DISTN(5),DISTX(5),PHOTO(20),IH(5)
     1 ,IK(5),IL(5)
      AMTOL=0.05
      PAMTOL=AMTOL*100.
      ANGTOL=3.0
      DPR=57.295780
   12 PI180=3.1415926535897932/180.
C
C     ****READ DETAILS OF DIFFRACTION PATTERN
C
    2 READR=5
      PRINTR=6
      WRITE (6,533)

   11 READ(5,101)(PHOTO(I),I=1,20)

      READ(READR,102) CAMCO,TOL,IOUT
      READ(READR,103)(DIST(N),N=1,5 )
      IF(DIST(4).EQ.0.0) GO TO 120
  110 LE=5
      GO TO 130
  120 LE=3
  130 LES=LE-1
      READ(READR,106)(ANGLES(MM),MM=1,4)
      DO 135 MM=1,LES
      COSANG=COS(ANGLES(MM)/DPR)
  135 ANGLES(MM)=(ARCOS(COSANG))*DPR
    3 DO 5 N=1,5
      IF(TOL-(DIST(N)*AMTOL)) 14,4,4
   14 DISTN(N)=DIST(N)-(DIST(N)*AMTOL)
      DISTX(N)=DIST(N)+(DIST(N)*AMTOL)
      GO TO 5
    4 DISTN(N)=DIST(N)-TOL
      DISTX(N)=DIST(N)+TOL
    5 CONTINUE
      WRITE(6,201) PHOTO
      WRITE (PRINTR,221)
      WRITE (PRINTR,221)
      WRITE (PRINTR,221)
      WRITE(PRINTR,202) CAMCO
      WRITE(6,203) TOL,PAMTOL
      WRITE(PRINTR,204)(DIST(K),K=1,5)
      WRITE(PRINTR,219) ANGTOL
      WRITE(PRINTR,220) (ANGLES(MM),MM=1,4)
C
C     ****READ DETAILS OF MATERIALS
C
   10 KONST=1
      READ(5,107)SYM,(TITLE(J),J=1,19)
      SYMI=SYM+1
      READ(READR,104) A,B,C,ALPHA,BETA,GAMMA
      WRITE(PRINTR,205) TITLE
      WRITE(6,221)
      WRITE(PRINTR,206)
      WRITE(PRINTR,207) A,B,C,ALPHA,BETA,GAMMA
C
C     ****COMPUTE RECIPROCAL CONSTANTS
C
      SINA=DSIN(ALPHA*PI180)
      SINB=DSIN(BETA*PI180)
      SING=DSIN(GAMMA*PI180)
      COSA=DCOS(ALPHA*PI180)
      COSB=DCOS(BETA*PI180)
      COSG=DCOS(GAMMA*PI180)
      V=1./(A*B*C*DSQRT(1.+2.*COSA*COSB*COSG-COSA*COSA-COSB*COSB-COSG*CO
     1SG))
      ASTAR=B*C*V*SINA
      BSTAR=A*C*V*SINB
      CSTAR=A*B*V*SING
      COSAS=(COSB*COSG-COSA)/(SINB*SING)
      COSBS=(COSA*COSG-COSB)/(SINA*SING)
      COSGS=(COSA*COSB-COSG)/(SINA*SINB)
      A11=ASTAR*ASTAR
```

```
      A22=BSTAR*BSTAR
      A33=CSTAR*CSTAR
      A12=2.*ASTAR*BSTAR*COSGS
      A13=2.*ASTAR*CSTAR*COSBS
      A23=2.*BSTAR*CSTAR*COSAS
C
C     ****DETERMINE LIMITS OF INDICES
C
    6 DISTMX=DISTX(1)
    7 DO 9 N=2,5
    8 IF(DISTMX.GT.DISTX(N)) GO TO 9
      DISTMX=DISTX(N)
    9 CONTINUE
      DMIN=CAMCO/(2.*DISTMX)
      DMIN=DMIN*DMIN
      DMAX=DSQRT(1/DMIN)
      HMAX=(DMAX/(ASTAR*DSQRT(1.-COSBS*COSBS)*SING))+1.
      KMAX=(DMAX/(BSTAR*DSQRT(1.-COSGS*COSGS)*SINA))+1.
      LMAX=(DMAX/(CSTAR*DSQRT(1.-COSAS*COSAS)*SINB))+1.
  503 WRITE(PRINTR,532)HMAX,KMAX,LMAX
C
C     ****GENERATE RECIPROCAL VECTORS
C
      PMAX=0
      RMAX=0
      TMAX=0
      LIST3=0
      NHMAX=2*HMAX+1
      NKMAX=2*KMAX+1
      NLMAX=2*LMAX+1
      DO 72 NH=1,NHMAX
      H=(HMAX+1)-NH
      DO 72 NK=1,NKMAX
      K=(KMAX+1)-NK
      DO 72 NL=1,NLMAX
      L=(LMAX+1)-NL
      IF(H.EQ.0.AND.K.EQ.0.AND.L.EQ.0) GO TO 15
      CALL PROHIB (SYMI,H,K,L,INC)
      IF (INC-1) 17,70,70
   15 LIST3=1
      GO TO 72
   17 DSQ=H*H*A11+H*K*A12+H*L*A13+K*K*A22+K*L*A23+L*L*A33
      D=SNGL(DSQRT(1./DSQ))
      RADI=CAMCO/(2.*D)
      DO 70 N=1,5,2
   18 IF(RADI-DISTN(N)) 70,19,19
   19 IF(RADI-DISTX(N)) 20,20,70
   20 IF(N-1) 40,21,22
   21 PMAX=PMAX+1
      M=PMAX
      GO TO 40
   22 IF(N-3) 40,25,26
   25 IF(LIST3.EQ.1) GO TO 70
      RMAX=RMAX+1
      M=RMAX
      GO TO 40
   26 TMAX=TMAX+1
      M=TMAX
   40 NN=(N+1)/2
      IND(NN,1,M)=H
      IND(NN,2,M)=K
      IND(NN,3,M)=L
      RAD(NN,M)=D
      IF(PMAX.GT.800.OR.RMAX.GT.800.OR.TMAX.GT.800) GO TO 1003
   70 CONTINUE
   72 CONTINUE
      GO TO 71
 1003 WRITE(PRINTR,218)
      GO TO 1004
   71 WRITE(PRINTR,208)PMAX,RMAX,TMAX
      GO TO (38,31,32,33,34,35,36,37),SYMI
   31 WRITE(6,534)
      GO TO 38
   32 WRITE(6,535)
      GO TO 38
   33 WRITE(6,536)
```

```
      GO TO 38
   34 WRITE(6,537)
      GO TO 38
   35 WRITE(6,538)
      GO TO 38
   36 WRITE(6,539)
      GO TO 38
   37 WRITE(6,540)
   38 IF(DIST(4)) 43,43,44
   43 IF(PMAX.EQ.0.OR.RMAX.EQ.0) GO TO 1007
   44 IF(PMAX.EQ.0.OR.RMAX.EQ.0.OR.TMAX.EQ.0) GO TO 1007
C
C     ****DETERMINE INDICES OF POINT '2'
C
   92 DO 1002 I=1,PMAX
   95 DO 1002 K=1,RMAX
      ZHB=(IND(1,1,I)+IND(2,1,K))/2.
      NHB=ZHB
      AZHB=NHB
      IF(ZHB-AZHB) 1002,96,1002
   96 ZKB=(IND(1,2,I)+IND(2,2,K))/2.
      NKB=ZKB
      AZKB=NKB
      IF(ZKB-AZKB) 1002,97,1002
   97 ZLB=(IND(1,3,I)+IND(2,3,K))/2.
      NLB=ZLB
      AZLB=NLB
      IF(ZLB-AZLB) 1002,98,1002
   98 IF(NHB.EQ.0.AND.NKB.EQ.0.AND.NLB.EQ.0) GO TO 1002
      CALL PROHIB (SYMI,NHB,NKB,NLB,INC)
      IF(INC-1) 99,1002,1002
C
C     ****CHECK POINT '2' IS WITHIN LIMITS OF RADIUS
C
   99 DSQ2=NHB*NHB*A11+NHB*NKB*A12+NHB*NLB*A13+NKB*NKB*A22+NKB*NLB*A23+
     1NLB*NLB*A33
      D2=SNGL(DSQRT(1./DSQ2))
      RADI2=CAMCO/(2.*D2)
      IF(RADI2.GE.DISTN(2).AND.RADI2.LE.DISTX(2)) GO TO 151
      GO TO 1002
C
C     ****DETERMINE INDICES OF POINT '4'
C
  151 IF(DIST(4).EQ.0.0) GO TO 159
      DO 1001 M=1,TMAX
      ZHD=(IND(2,1,K)+IND(3,1,M))/2.
      NHD=ZHD
      AZHD=NHD
      IF(ZHD-AZHD) 1001,153,1001
  153 ZKD=(IND(2,2,K)+IND(3,2,M))/2.
      NKD=ZKD
      AZKD=NKD
      IF(ZKD-AZKD) 1001,154,1001
  154 ZLD=(IND(2,3,K)+IND(3,3,M))/2.
      NLD=ZLD
      AZLD=NLD
      IF(ZLD-AZLD) 1001,155,1001
  155 IF(NHD.EQ.0.AND.NKD.EQ.0.AND.NLD.EQ.0) GO TO 1001
      CALL PROHIB (SYMI,NHD,NKD,NLD,INC)
      IF (INC-1) 156,1001,1001
C
C     ****CHECK POINT '4' IS WITHIN LIMITS OF RADIUS
C
  156 DSQ4=NHD*NHD*A11+NHD*NKD*A12+NHD*NLD*A13+NKD*NKD*A22+NKD*NLD*A23+
     1NLD*NLD*A33
      D4=SNGL(DSQRT(1./DSQ4))
      RADI4=CAMCO/(2.*D4)
      IF(RADI4.GE.DISTN(4).AND.RADI4.LE.DISTX(4)) GO TO 158
      GO TO 1001
C
C     ****SORT PLANES INTO ARRAYS
C
  159 L=1
      M=1
  158 IH(1)=IND(1,1,I)
      IH(2)=NHB
```

```
      IH(3)=IND(2,1,K)
      IH(4)=NHD
      IH(5)=IND(3,1,M)
      IK(1)=IND(1,2,I)
      IK(2)=NKB
      IK(3)=IND(2,2,K)
      IK(4)=NKD
      IK(5)=IND(3,2,M)
      IL(1)=IND(1,3,I)
      IL(2)=NLB
      IL(3)=IND(2,3,K)
      IL(4)=NLD
      IL(5)=IND(3,3,M)
C
C     ****CALCULATE ANGLES BETWEEN PLANES
C
      MM=0
      II=1
  160 IF(DIST(4)) 161,161,162
  161 NZ=3
      GO TO 163
  162 NZ=5
  163 DO 170 NN=2,NZ
      S=((IH(II)*IH(NN)*ASTAR*ASTAR)+(IK(II)*IK(NN)*BSTAR*BSTAR)+(IL(II)
     1*IL(NN)*CSTAR*CSTAR)+((IK(II)*IL(NN)+IL(II)*IK(NN))*BSTAR*CSTAR*CO
     2SAS)+((IL(II)*IH(NN)+IL(NN)*IH(II))*CSTAR*ASTAR*COSBS)+((IH(II)*IK
     3(NN)+IK(II)*IH(NN))*ASTAR*BSTAR*COSGS))
      T=((IH(II)*IH(II)*ASTAR*ASTAR)+(IK(II)*IK(II)*BSTAR*BSTAR)+(IL(II)
     1*IL(II)*CSTAR*CSTAR)+(2.0*IH(II)*IK(II)*ASTAR*BSTAR*COSGS)+(2.0*IL
     2(II)*IH(II)*CSTAR*ASTAR*COSBS)+(2.0*IK(II)*IL(II)*BSTAR*CSTAR*COSA
     3S))
      U=((IH(NN)*IH(NN)*ASTAR*ASTAR)+(IK(NN)*IK(NN)*BSTAR*BSTAR)+(IL(NN)
     1*IL(NN)*CSTAR*CSTAR)+(2.0*IH(NN)*IK(NN)*ASTAR*BSTAR*COSGS)+(2.0*IL
     2(NN)*IH(NN)*CSTAR*ASTAR*COSBS)+(2.0*IK(NN)*IL(NN)*BSTAR*CSTAR*COSA
     3S))
      W=SQRT(T*U)
      YY=ARCOS(S/W)
      YY=YY*DPR
      MM=MM+1
      IF(YY.GT.(ANGLES(MM)+ANGTOL).OR.YY.LT.(ANGLES(MM)-ANGTOL)) GO TO 1
     1001
      ANGACT (MM)=YY
  170 CONTINUE
      CALL OARRAY(KONST,IH,IK,IL,I,J,K,L,M,PI180)
 1001 CONTINUE
 1002 CONTINUE
 1007 KONST=3
      CALL OARRAY(KONST,IH,IK,IL,I,J,K,L,M,PI180)
 1004 READ(READR,105) ICARD
      IF (ICARD-1.) 11,10,11
C
C     **** FORMAT STATEMENTS ----INPUT
C
  101 FORMAT(20A4)
  102 FORMAT(F8.4,F6.3,I1)
  103 FORMAT(5F8.2)
  104 FORMAT(3F10.3,3F10.2)
  105 FORMAT(I1)
  106 FORMAT(4F8.2)
  107 FORMAT(I1,19A4)
C
C     ****FORMAT STATEMENTS ---- OUTPUT
C
  201 FORMAT(1H1,20A4)
  202 FORMAT(1H0,'CAMERA CONSTANT=',F8.4)
  203 FORMAT(1H0,'POSITION TOLERANCE =',F6.3,' MILLIMETRES  (MINIMUM OVE
     1R-RIDING TOLERANCE OF + OR - ',F3.1,' PERCENT OF DIFF. DISTANCE PR
     2EVAILS)')
  204 FORMAT(1H0,'DIFFRACTED DISTANCES OF SPOTS ARE',//5F10.2,'   (MILLI
     1METRES)')
  205 FORMAT(1H0,///////1X,19A4)
  206 FORMAT(1H0,20X,'REAL CELL CONSTANTS'//26X,'A',8X,'B',8X,'C',6X,'AL
     1PHA',5X,'BETA',4X,'GAMMA'/)
  207 FORMAT(20X,6F9.3)
```

```
  208 FORMAT(1H0,20X,'MAXIMUM DIMENSIONS OF ARRAYS ARE  ',5I6)
  218 FORMAT(1H0,'ARRAY LIMITS EXCEEDED---MATERIAL CHECK CANCELLED')
  219 FORMAT(1H0,'ANGLE TOLERANCE = ',F5.2,' DEGREES')
  220 FORMAT(1H0,'MEASURED ANGLES BETWEEN SPOTS ARE',// 4F10.2)
  221 FORMAT(55H***********************************************************)
  532 FORMAT(1H0,20X,19HMAXIMUM INDICES ARE,3I4)
  533 FORMAT(1H0,6X,30HPROGRAM CRYSTAL IDENTIFICATION)
  534 FORMAT(1H0,20X,'PROHIBITED REFLECTIONS FOR THE FACE CENTRED CELL T
     1YPE F HAVE BEEN OMITTED')
  535 FORMAT(1H0,20X,'PROHIBITED REFLECTIONS FOR THE BODY CENTRED CELL T
     1YPE I HAVE BEEN OMITTED')
  536 FORMAT(1H0,20X,'PROHIBITED REFLECTIONS FOR THE FACE CENTRED CELL T
     1YPE A HAVE BEEN OMITTED')
  537 FORMAT(1H0,20X,'PROHIBITED REFLECTIONS FOR THE FACE CENTRED CELL T
     1YPE B HAVE BEEN OMITTED')
  538 FORMAT(1H0,20X,'PROHIBITED REFLECTIONS FOR THE FACE CENTRED CELL T
     1YPE C HAVE BEEN OMITTED')
  539 FORMAT(1H0,20X,'PROHIBITED REFLECTIONS FOR THE OBVERSE RHOMBOHEDRO
     1N (HEXAGONAL CELL) TYPE R HAVE BEEN OMITTED')
  540 FORMAT(1H0,20X,'PROHIBITED REFLECTIONS FOR THE REVERSE RHOMBOHEDRO
     1N (HEXAGONAL CELL) TYPE R HAVE BEEN OMITTED')
      END
```

```
      SUBROUTINE PROHIB (SYMI,H,K,L,INC)
C
C     TEST INDICES AND ELIMINATE PROHIBITED REFLECTIONS
C
      INTEGER H,SYMI
      GO TO (68,13,15,61,62,13,63,64),SYMI
   13 AKH=(K+H)/2.
      KH=AKH
      BKH=KH
      IF(AKH.NE.BKH) GO TO 70
      IF(SYMI.NE.2) GO TO 68
   62 ALH=(L+H)/2.
      LH=ALH
      BLH=LH
      IF(ALH.NE.BLH) GO TO 70
      IF(SYMI.NE.2) GO TO 68
   61 AKL=(K+L)/2.
      KL=AKL
      BKL=KL
      IF(AKL.NE.BKL) GO TO 70
      GO TO 68
   15 AKHL=(H+K+L)/2.
      GO TO 66
   63 AKHL=(K+L-H)/3.
      GO TO 66
   64 AKHL=(H+L-K)/3.
   66 KHL=AKHL
      BKHL=KHL
      IF(AKHL.NE.BKHL) GO TO 70
   68 INC=0
   69 GO TO 75
   70 INC=1
   75 RETURN
      END
```

```
      SUBROUTINE OARRAY(KONST,IH,IK,IL,IA,JA,KA,LA,MA,PI180)
C
C     SUBROUTINE 'OARRAY' SORTS SOLUTIONS AND PRINTS RESULTS
C
      COMMON RADIX(5,50),ANGIX(4,50),IZONES(3,50,50),IHP(5,50,50),
     1IKP(5,50,50),ILP(5,50,50),KSETS,KP(50),RAD(3,801),KS,ANGLES(4),
     2IZOS(3,50),IOUT,DIST(5),DISTIX(5,50),ANGACT(4),CAMKO(50),CAMCO,
     3LE,LES,ADEVN(50),DEVSQ(50),KT,D2,D4
      DIMENSION IH(5),IK(5),IL(5),ANG(5),
     1IZON(3),IZO(3),DIV(3),ADIV(3),JDIV(3),DEVN(5),DEV(50),HSUM(50)
      DOUBLE PRECISION PI180
      IF(KONST-3) 1030,600,1030
C
C     CALCULATE LOWEST ORDER ZONE AXES
C
 1030 IZON(1)=IK(1)*IL(2)-IK(2)*IL(1)
 1031 IZON(2)=IL(1)*IH(2)-IL(2)*IH(1)
 1032 IZON(3)=IH(1)*IK(2)-IH(2)*IK(1)
      DO 1033 I=1,3
      IZO(I)=IZON(I)
 1033 CONTINUE
 1040 IF(IABS(IZO(1))-IABS(IZO(2)))1041,1042,1042
 1041 INV=IZO(2)
      IZO(2)=IZO(1)
      IZO(1)=INV
 1042 IF(IABS(IZO(2))-IABS(IZO(3)))1043,1045,1045
 1043 INV=IZO(3)
      IZO(3)=IZO(2)
      IZO(2)=INV
      GO TO 1040
 1045 INCMAX=IABS(IZO(1))
 1083 DO 1061 INC=1,INCMAX
      DO 1060 IZ=1,3
 1046 DIV(IZ)=IABS(IZO(IZ))/((INCMAX+1.)-INC)
      JDIV(IZ)=DIV(IZ)+0.001
      ADIV(IZ)=JDIV(IZ)
 1060 CONTINUE
      IF(DIV(1).EQ.ADIV(1).AND.DIV(2).EQ.ADIV(2).AND.DIV(3).EQ.ADIV(3))G
     1O TO 1063
 1061 CONTINUE
 1063 DO 1070 I=1,3
      IF(IZON(I))64,65,65
   64 IZON(I)=IZON(I)/((INCMAX+1)-INC)-0.001
      GO TO 1070
   65 IZON(I)=IZON(I)/((INCMAX+1)-INC)+0.001
 1070 CONTINUE
C
C     LOAD SOLUTIONS INTO OUTPUT ARRAYS
C
      GO TO (100,400,600,400) KONST
  100 KONST=2
      DO 110 I=1,50
  110 KP(I)=0
      KT=0
      KSETS=0
  200 IF(KT.LT.50) GO TO 208
  900 KONST=4
      GO TO 350
  208 KT=KT+1
      KSETS=KSETS+1
      RADIX(1,KT)=RAD(1,IA)
      RADIX(2,KT)=D2
      RADIX(3,KT)=RAD(2,KA)
      RADIX(4,KT)=D4
      RADIX(5,KT)=RAD(3,MA)
      DO 210 I=1,4
  210 ANGIX(I,KT)=ANGACT(I)
      DO 220 I=1,5
      DISTIX(I,KT)=DIST(I)
  220 CONTINUE
      KS=KT
  300 KP(KS)=KP(KS)+1
```

```
C
C     CALCULATE MEAN DEVIATION OF SPOTS FROM TRUE POSITIONS
C
      DEV(KS)=0.
      DEVSQ(KS)=0.
      RADTOT=0.
      DISTOT=0.
   30 DO 50 N=1,LE
      RADTOT=RADTOT+RADIX(N,KS)
   50 DISTOT=DISTOT+(CAMCO/(2*DISTIX(N,KS)))
      CAMINC=((RADTOT-DISTOT)/DISTOT)+1.
      CAMKO(KS)=CAMCO*CAMINC
      ANGDEV=0.
      DO 55 N=1,LES
   55 ANGDEV=ANGDEV+(ANGIX(N,KS)-ANGLES(N))
      ANGDE=ANGDEV/LE
      ANG(1)=0.-ANGDE
      DO 56 N=1,LES
      M=N+1
   56 ANG(M)=(ANGIX(N,KS)-ANGLES(N))-ANGDE
      DO 58 N=1,LE
      X1=DISTIX(N,KS)-((CAMKO(KS)/(2.*RADIX(N,KS)))*DCOS(ANG(N)*PI180))
      X2=(CAMKO(KS)/(2.*RADIX(N,KS)))*DSIN(ANG(N)*PI180)
   58 DEVSQ(KS)=DEVSQ(KS)+SQRT((X1*X1)+(X2*X2))
      IF(KP(KS).GT.50) GO TO 1000
      K=KP(KS)
      DO 310 I=1,3
      IZONES(I,K,KS)=IZON(I)
  310 IZOS(I,KS)=JDIV(I)
      DO 320 I=1,5
      IHP(I,K,KS)=IH(I)
      IKP(I,K,KS)=IK(I)
  320 ILP(I,K,KS)=IL(I)
      GO TO 1000
C
C     CHECK FOR SYMMETRICAL EQUIVALENT SOLUTIONS
C
  400 DO 410 KS=1,KT
      IF(RAD(1,IA).EQ.RADIX(1,KS).AND.   D2    .EQ.RADIX(2,KS).AND.RAD(2
     1,KA).EQ.RADIX(3,KS).AND.   D4    .EQ.RADIX(4,KS).AND.RAD(3,MA).EQ.
     2RADIX(5,KS)) GO TO 420
      GO TO 410
  420 DO 430 L=1,4
      IF(ANGIX(L,KS).NE.ANGACT(L)) GO TO 410
  430 CONTINUE
      DO 440 I=1,3
      IF(IZOS(I,KS).NE.JDIV(I)) GO TO 410
  440 CONTINUE
      GO TO 300
  410 CONTINUE
      GO TO 200
  600 IF(KSETS) 1600,1600,350
C
C     SORT SETS OF RESULTS INTO ORDER OF ACCURACY
C
  350 N=KT
      M=N
 1120 M=M/2
      IF(M) 1130,1140,1130
 1130 K=N-M
      J=1
 1141 I=J
 1149 L=I+M
      IF(DEVSQ(I)-DEVSQ(L)) 1160,1160,1150
 1150 B=DEVSQ(I)
      DEVSQ(I)=DEVSQ(L)
      DEVSQ(L)=B
      CAM=CAMKO(I)
      CAMKO(I)=CAMKO(L)
      CAMKO(L)=CAM
      DO 1153 NM=1,5
      AR=RADIX(NM,I)
      RADIX(NM,I)=RADIX(NM,L)
      RADIX(NM,L)=AR
```

```
      AD=DISTIX(NM,I)
      DISTIX(NM,I)=DISTIX(NM,L)
      DISTIX(NM,L)=AD
      DO 1153 KK=1,50
      IP=IHP(NM,KK,I)
      IHP(NM,KK,I)=IHP(NM,KK,L)
      IHP(NM,KK,L)=IP
      IP=IKP(NM,KK,I)
      IKP(NM,KK,I)=IKP(NM,KK,L)
      IKP(NM,KK,L)=IP
      IP=ILP(NM,KK,I)
      ILP(NM,KK,I)=ILP(NM,KK,L)
 1153 ILP(NM,KK,L)=IP
      DO 1154 NN=1,3
      IQ=IZOS(NN,I)
      IZOS(NN,I)=IZOS(NN,L)
      IZOS(NN,L)=IQ
      DO 1154 KK=1,50
      IP=IZONES(NN,KK,I)
      IZONES(NN,KK,I)=IZONES(NN,KK,L)
 1154 IZONES(NN,KK,L)=IP
      DO 1155 NA=1,4
      AA=ANGIX(NA,I)
      ANGIX(NA,I)=ANGIX(NA,L)
 1155 ANGIX(NA,L)=AA
      KZ=KP(I)
      KP(I)=KP(L)
      KP(L)=KZ
      I=(I-M)
      IF(I-1) 1160,1149,1149
 1160 J=J+1
      IF(J-K) 1141,1141,1120
 1140 IF (KONST.NE.4) GO TO 1180
      KT=30
      DO 1165 LS=31,50
 1165 KP(LS)=0
      GO TO 208
C
C     SORT ZONEAXES WITHIN EACH SET INTO ORDER
C
 1180 IF(KT.LE.30) GO TO 1185
      KT=30
 1185 DO 1300 KS=1,KT
      N=KP(KS)
      DO 1210 KK=1,N
 1210 HSUM(KK)=IZONES(1,KK,KS)+IZONES(2,KK,KS)+IZONES(3,KK,KS)
      M=N
 1220 M=M/2
      IF(M) 1230,1240,1230
 1230 K=N-M
      J=1
 1241 I=J
 1249 L=I+M
      IF(HSUM(L)-HSUM(I)) 1260,1260,1250
 1250 HB=HSUM(I)
      HSUM(I)=HSUM(L)
      HSUM(L)=HB
      DO 1252 NN=1,3
      IP=IZONES(NN,I,KS)
      IZONES(NN,I,KS)=IZONES(NN,L,KS)
 1252 IZONES(NN,L,KS)=IP
      DO 1254 NN=1,5
      IP=IHP(NN,I,KS)
      IHP(NN,I,KS)=IHP(NN,L,KS)
      IHP(NN,L,KS)=IP
      IP=IKP(NN,I,KS)
      IKP(NN,I,KS)=IKP(NN,L,KS)
      IKP(NN,L,KS)=IP
      IP=ILP(NN,I,KS)
      ILP(NN,I,KS)=ILP(NN,L,KS)
 1254 ILP(NN,L,KS)=IP
      I=(I-M)
      IF(I-1)1260,1249,1249
 1260 J=J+1
      IF(J-K)1241,1241,1220
 1240 DO 1293 IPZ=1,N
```

```
      IPZN=IPZ-1
      IF (IPZ.EQ.1) GO TO 1242
      IF(HSUM(IPZ).LT.HSUM(IPZN)) GO TO 1300
 1242 IF(IZONES(1,IPZ,KS).LE.IZONES(2,IPZ,KS).AND.
     1IZONES(2,IPZ,KS).LE.IZONES(3,IPZ,KS))GO TO 1290
      GO TO 1293
 1290 DO 1291 NN=1,3
      IP=IZONES(NN,IPZ,KS)
      IZONES(NN,IPZ,KS)=IZONES(NN,1,KS)
 1291 IZONES(NN,1,KS)=IP
      DO 1292 NN=1,5
      IP=IHP(NN,IPZ,KS)
      IHP(NN,IPZ,KS)=IHP(NN,1,KS)
      IHP(NN,1,KS)=IP
      IP=IKP(NN,IPZ,KS)
      IKP(NN,IPZ,KS)=IKP(NN,1,KS)
      IKP(NN,1,KS)=IP
      IP=ILP(NN,IPZ,KS)
      ILP(NN,IPZ,KS)=ILP(NN,1,KS)
 1292 ILP(NN,1,KS)=IP
 1293 CONTINUE
 1300 CONTINUE
C
C     WRITE OUT RESULTS
C
      IF(KSETS-1) 1295,1295,1294
 1294 WRITE(6,1001) KSETS
 1295 DO 1500 KS=1,KT
      IF(KP(KS)-1)1302,1302,1301
 1301 WRITE(6,1002)KS,(IZONES(NN,1,KS),NN=1,3),KS,KP(KS)
      WRITE(6,1016)
      GO TO 1305
 1302 WRITE(6,1003)KS,(IZONES(NN,1,KS),NN=1,3)
      WRITE(6,1016)
 1305 WRITE(6,1004)
      DO 1308 I=1,LE
      DISTIX(I,KS)=CAMKO(KS)/(2.*DISTIX(I,KS))
 1308 WRITE(6,1005)I,IHP(I,1,KS),IKP(I,1,KS),ILP(I,1,KS),RADIX(I,KS),
     1DISTIX(I,KS)
      WRITE(6,1008)CAMKO(KS)
      WRITE(6,1009)CAMCO
      ADEVN(KS)=DEVSQ(KS)/LE
      WRITE(6,1010)ADEVN(KS)
      WRITE(6,1007)
      DO 1390 I=2,LE
      LES=I-1
 1390 WRITE(6,1006)I,ANGIX(LES,KS),ANGLES(LES)
      IF(IOUT-1)1500,1395,1500
 1395 IF(KP(KS).LE.1) GO TO 1500
      WRITE(6,1011)KS
      KKMAX=KP(KS)
      IF(LE-3)1440,1396,1440
 1396 WRITE(6,1013)
      DO 1400 I=2,KKMAX
 1400 WRITE(6,1015)(IZONES(L,I,KS),L=1,3),IHP(1,I,KS),
     1IKP(1,I,KS),ILP(1,I,KS),IHP(2,I,KS),IKP(2,I,KS),
     2ILP(2,I,KS),IHP(3,I,KS),IKP(3,I,KS),ILP(3,I,KS)
      GO TO 1500
 1440 WRITE(6,1012)
      DO 1460 I=2,KKMAX
 1460 WRITE(6,1014)(IZONES(L,I,KS),L=1,3),IHP(1,I,KS),
     1IKP(1,I,KS),ILP(1,I,KS),IHP(2,I,KS),IKP(2,I,KS),
     2ILP(2,I,KS),IHP(3,I,KS),IKP(3,I,KS),ILP(3,I,KS),
     3IHP(4,I,KS),IKP(4,I,KS),ILP(4,I,KS),IHP(5,I,KS),
     4IKP(5,I,KS),ILP(5,I,KS)
 1500 CONTINUE
```

```
      KS=0
      KSETS=0
      GO TO 1000
 1600 WRITE(6,1017)
      GO TO 1000
 1000 RETURN
C
C     FORMAT STATEMENTS----OUTPUT
C
 1001 FORMAT(1H0,I3,' SETS OF POSSIBLE ZONE AXES INDEX WITHIN SPECIFIED
     1LIMITS')
 1002 FORMAT(1H0,/////11X,'SET ',I2,'    ZONE AXIS (',3I3,')',10X,'(SET
     1',I2,' HAS ',I2,' SYMMETRICAL EQUIVALENT SOLUTIONS)')
 1003 FORMAT(1H0,/////11X,'SET ',I2,'    ZONE AXIS (',3I3,')',10X,'(THER
     1E ARE NO SYMMETRICAL EQUIVALENT SOLUTIONS)')
 1004 FORMAT(1H0,20X,'POINT',10X,'PLANE',10X,'DSPACE',10X,'ESTIMATED ','
     1DSPACE FROM DIFF. PATTERN'/)
 1005 FORMAT(1H ,22X,I1,9X,'(',3I3,')',7X,F6.3,15X,F6.3)
 1006 FORMAT(1H ,20X,'ANGLE BETWEEN PLANES 1 & ',I1,
     1' = ',F6.2,'  MEASURED ',F6.2,' DEGREES')
 1007 FORMAT(1H )
 1008 FORMAT(1H0,20X,'N.B. BEST FIT CAMERA CONSTANT USED ',
     1IN ABOVE ESTIMATES OF D SPACINGS = ',F7.3)
 1009 FORMAT(1H ,65X,'(INPUT CAMERA CONSTANT = ',F7.3,')')
 1010 FORMAT(1H0,20X,'MEAN DEVIATION OF MEASURED SPOTS ',
     1FROM TRUE POSITIONS = ',F5.3,' MILLIMETRES')
 1011 FORMAT(1H0,20X,'SYMMETRICAL EQUIVALENT SOLUTIONS ',
     1FOR SET ',I2)
 1012 FORMAT(1H0,21X,'ZONE AXIS',10X,'POINT 1',10X,'POINT 2',
     110X,'POINT 3',10X,'POINT 4',10X,'POINT 5',/)
 1013 FORMAT(1H0,21X,'ZONE AXIS',10X,'POINT 1',10X,'POINT 2',
     110X,'POINT 3',/)
 1014 FORMAT(1H ,20X,'(',3I3,')',7X,'(',3I3,')',6X,'(',
     13I3,')',6X,'(',3I3,')',6X,'(',3I3,')',6X,'(3I3,')')
 1015 FORMAT(1H ,20X,'(',3I3,')',7X,'(',3I3,')',6X,'(',
     13I3,')',6X,'(',3I3,')')
 1016 FORMAT(1H ,10X,'******    **********************')
 1017 FORMAT(1H0,20X,'NO IDENTIFICATION',/21X,'******************')
      END
```